

ARBITRATION MECHANISM FOR PACKET TRANSMISSION

Related Applications

[0001] The present application is a continuation of co-pending U.S. Patent Application Serial No. 09/411,429 entitled ARBITRATION MECHANISM FOR PACKET TRANSMISSION filed on October 1, 1999.

Field of the Invention

[0002] The present invention relates to an arbitration mechanism for packet transmission and is particularly but not exclusively concerned with packet transmission in a packet transmission network.

Background to the Invention

[0003] Computer systems and integrated circuit processors exist which implement transactions with the dispatch and receipts of packets. Request packets define an operation to be performed and response packets indicate that a request has been received. The integrated circuit processor can comprise a plurality of functional modules connected to a packet router for transmitting and receiving the request and response packets. In such a system, it is necessary to arbitrate between requests received from the functional modules for controlling the flow of packets on the packet router (whether request or response packets). Typically, multiple functional modules are able to access the same bus or the same part of the system which leads to competition for that bus or for the same destination or target modules. The complexity of the arbitration mechanism is a function of the number of devices in the system and the arbitration algorithm which is used.

[0004] As the size and clock frequency of such systems, in particular integrated systems, increases, and the complexity of the arbitration function increases, the time required to make an arbitration decision for accesses to resources such as the system bus or target modules becomes critical. In many systems, it is not possible to make an arbitration decision and effect a packet transfer in a single clock cycle. This leads to a drop in potential performance as gaps are inserted increasing the latency between a request for packet transfer being made and the packet transfer actually being implemented.

[0005] It is an aim of the present invention to reduce the impact of arbitration decision latency on a system.

Summary of the Invention

[0006] According to one aspect of the present invention there is provided a computer system comprising: a plurality of functional modules interconnected via a packet router, each functional module having packet handling circuitry for generating and receiving packets conveyed by the packet router; and a routing control mechanism for controlling the flow of packets on the packet router, said routing control mechanism being connected to said functional modules and to said packet router, wherein each functional module is operable to generate to the routing control mechanism a transfer request to request transfer of a current packet and an arbitration request with a destination indicator identifying a destination of a later packet and wherein the routing control mechanism is operable to accept said arbitration request with the destination indicator of the later packet and to effect a routing decision relating to the arbitration request while implementing the transfer of the current packet requested by the transfer request.

[0007] In the described embodiment, the arbitration request relates to a packet issued two cycles after the current packet by the same functional module. That is, the arbitration unit effectively hides a latency of two decision cycles. However, a latency period of at least one cycle still represents a considerable advantage. That is, transfer of a current packet can be implemented while a subsequent arbitration decision is effected. There can be any suitable number of cycles in the latency period.

[0008] At least a set of said functional modules can act as initiator modules for generating request packets for implementing transactions, each request packet including the destination indicator. A further set of the functional modules can act as target modules for receiving said request packets and for issuing respective response packets. The response packets can be arbitrated in the same way as request packets with associated response transfer and response arbitration request signals. Each functional module can be operable to generate a grant signal indicating that it is in a state to receive a packet. That grant signal is used by the routing control mechanism to effect packet transfers.

[0009] The routing control mechanism is preferably operable to issue an arbitration grant signal indicating that it has committed an arbitration decision to await transfer. This acts as a handshake mechanism to indicate to the functional modules that an arbitration request has been granted and the system is committed to transferring that packet. This allows the functional module to begin negotiation for an arbitration decision for a subsequent packet, possibly before the transfer of the previous packet.

[0010] In the preferred embodiment, each packet has an address field identifying a destination module using the destination indicator and identifying an address location within the target module to which the packet is to be directed. The arbitration request is associated with the destination indicator to allow arbitration decisions to be made while the transfer request is associated with the location within the target module.

[0011] Another aspect of the invention provides a functional module for connection to a packet router and having packet handling circuitry for generating and receiving packets conveyed by the packet router, the functional module being operable to generate packet flow control signals including a transfer request requesting transfer of a current packet and an arbitration request signal with a destination indicator identifying a destination of a later packet, wherein the arbitration request signal is issued when the later packet is ready for transfer.

[0012] Another aspect of the invention provides a routing control mechanism for routing packets between a plurality of functional modules interconnected via a packet router, the routing mechanism comprising: an arbitration mechanism for effecting routing decisions for each packet for which a request signal is received; a decision queue for holding at least one decision which has been made by the arbitration mechanism; and a transfer mechanism for transferring packets relating to queued decisions to the packet router to implement a packet transfer, wherein the arbitration mechanism is operable to effect a routing decision for a later packet while a current packet is being transferred by the transfer mechanism onto the packet router.

[0013] Another aspect of the invention provides a plurality of functional modules interconnected via a packet router, each functional module having packet handling circuitry for generating and receiving packets conveyed by the packet router; and a routing control mechanism for controlling the flow of packets on the packet router, said routing control mechanism being connected to said functional modules and to said packet router, the routing control mechanism comprising: an arbitration mechanism for effecting routing decisions for each packet for which a request signal is received; a queue for holding at least one routing decision which has been made by the arbitration mechanism; and a transfer mechanism for transferring packets relating to queued decisions to the packet router to implement a packet transfer, wherein the arbitration mechanism is operable to effect a routing decision for a later packet while a current packet is being transferred by the transfer mechanism onto the packet router.

[0014] Another aspect of the invention provides a method of effecting pipelined routing decisions in an integrated circuit comprising a plurality of functional modules interconnected via a packet router wherein each functional module generates a transfer request for a current packet and an arbitration request with a destination indicator of a later packet, the method comprising: effecting a transfer of the current packet based on an earlier routing control decision while making a routing control decision in relation to the later packet.

[0015] For a better understanding of the present invention and to show how the same may be carried into effect reference will now be made by way of example to the accompanying drawings.

Brief Description of the Drawings

- [0016] Figure 1 is a block diagram of an integrated computer system;
- [0017] Figure 2 is a schematic diagram illustrating the connections of a functional module to an arbitration unit;
- [0018] Figure 3 is a schematic block diagram illustrating the arbitration control signals;
- [0019] Figures 4 and 5 illustrate the format of packets routed by the packet router;
- [0020] Figure 6 is a schematic block diagram of the arbitration unit; and
- [0021] Figures 7a and 7b are timing diagrams illustrating operation of the arbitration unit.

Description of the Preferred Embodiment

[0022] Figure 1 illustrates an integrated circuit according to an embodiment of the invention. On each chip 11 a central processor unit (CPU) 12 is connected to a plurality of modules M by a data and address path 15 arranged to carry bit packets in parallel form. The modules as well as the CPU unit 12 each include packet handling circuitry 2 used in the generation and receipt of bit packets on the path 15. The path 15 is referred to herein as a packet router or routing bus. Two main types of packet are used on the data and address path 15, each including a destination indicator or address to indicate the required destination module connected to the path 15. The packets include request packets which are generated by an initiator module and response packets which are generated by a target module. A module may act as both an initiator and a target. Response packets are of two types, ordinary responses or error responses. These are discussed in more detail later.

The modules M as well as the CPU unit 12 each have packet handling circuitry 2 for handling packet formation and receipt of requests, ordinary responses and error responses.

[0023] As shown most clearly in Figure 2, the routing bus 15 provides bi-directional connections to each module. In this example the bus consists of parallel request and response buses 7, 9 and a dedicated control bus 11 provided respectively for each module so as to link the modules to an arbitration unit 22. Each module is connected to the routing bus via a port 4 and is provided with an interface 6 incorporating a state machine so as to interchange control signals and data between the port 4 and the interface 6.

[0024] Figure 3 is a block diagram illustrating relevant functional components of the chip of Figure 1 to illustrate the concept of targets and initiator modules. It also illustrates the control signals on the control bus 11 for initiator and target modules. The modules are labeled M1, M2, M3 and M4 and may include any of the modules M described with reference to Figure 1. Modules M1 and M2 both have target and initiator functions as illustrated by the separate target and initiator parts of the interface 6 of each module. Module M3 acts only as an initiator and module M4 acts only as a target. Signals from the interfaces 6 are supplied to central control logic which forms part of the arbitration unit 22. The arbitration unit 22 issues request routing control signals and response routing controls to the routing bus network 15. The arbitration control signals are discussed more fully hereinafter.

[0025] In the example shown in Figure 1, the various modules M include a debug module 30 which includes an external link 31 for transmitting packets across the chip boundary (to,

for example, host 60), an external memory interface (EMI) 32 having an external bus connection 33 leading to an external memory 50, clock circuitry 34, various peripheral interfaces 35, a peripheral component interface (PCI) 37 with an external connection 38, a DMA unit 25 for effecting memory accesses as well as the arbitration unit 22. The CPU unit 12 includes a plurality of instruction execution units 40, a plurality of registers 41, and a cache 42. The CPU unit 12 also includes packet handling circuitry 2 connected to the execution units 40. The routing bus 15 is arranged to transmit to the modules M both request and response packets for effecting memory access transactions as discussed further herein. These packets may be generated by software as a result of instruction execution by a CPU or by hardware responsive to detection of a packet. The packets may be generated on-chip and distributed on the bus 15 or generated off-chip and supplied to the on-chip bus 15 through an external connection such as the link 31 associated with the debug module 30.

[0026] The CPU 12 can be operated in a conventional manner receiving instructions from a program memory and effecting data read or write operations with the cache 42 on-chip. Additionally external memory accesses for read or write operations may be made through the external memory interface 32 and bus connection 33 to the external memory 50. Each packet is constructed from a series of cells or tokens, the end of the packet being identified by an end of packet (eop) cell. Each packet cell comprises a number of fields which characterize the packet. Each packet is transmitted by a source module and is directed to a destination module. An initiator can issue request packets and act on response packets. A target can receive and act on requests and issue responses. Thus, a source module may be an initiator or a

target depending on the nature of the packet. The source module uses its associated port 4 to transmit a packet onto the routing bus 15. The routing bus 15 arranges for the packet to be routed to the port 4 associated with the destination module. The destination module then receives that packet from its associated port. The source and destination modules can be the same.

[0027] A transaction is an exchange of packets that allows a module to access the state of another module. A transaction comprises the transfer of a request packet from a source module to a destination module, followed by the transfer of a response packet from that destination module (now acting as a responding module) back to the source module which made the original request. The request packet initiates a transaction and its contents determine the access to be made. The response packet completes the transaction and its contents indicate the result of the access. A response packet also indicates whether the request was valid or not. If the request was valid, a so-called ordinary response packet is sent. If the request was invalid, an error response packet is transmitted. Some modules act only as initiators and thus their packet handling circuitry 2 is capable only of the generation of request packets. Some modules act only as targets (e.g., module M4 in Figure 3), and therefore their packet handling circuitry 2 is capable only of generating response packets. In that case, both ordinary responses and error responses can be generated. However, some modules are capable of acting both as initiators and as targets (e.g., modules M1 and M2 in Fig. 3), and their packet handling circuitry 2 is capable of generating both request and response type packets.

[0028] The format of the multi-bit packets used on the routing bus 15 in the microcomputer are illustrated by way of example in Figures 4 and 5. Figure 4 shows the information carried by each request cell. Figure 5 shows the information carried by each response cell. This information is conveyed onto the packet router 15 from the interfaces 6 as signals discussed more fully hereinafter.

[0029] Any appropriate arbitration algorithm can be used by the arbitration unit 22 to control access to the bus by initiator and target modules. The arbitration unit determines if more than one initiator is requesting access to a routing resource/bus required to access a particular target or group of targets, which port gains access to the target resources. This can be done using a fixed priority size algorithm or a more complex algorithm such as least recently used. The precise algorithm which is used can be determined by a person skilled in the art for the particular implementation. The arbitration decision is made using the following information: initiators requesting a particular target resource, target availability, position in message and the arbitration algorithm which is selected. The arbitration unit also arbitrates for a routing resource to return responses to the initiator based on the following information: the target requesting the resource, initiator availability, location in message and the selected arbitration algorithm.

[0030] According to the embodiment of the invention described herein, the arbitration unit 22 is also capable of making an arbitration decision about transfers to be effected after the current transfer. It will be appreciated that, in the absence of the additional signals discussed later with reference to the preferred embodiment of the invention, the arbitration unit would determine a sequence of transfers

responding to requests made to it. That is, it would queue requests for arbitration received from various modules attempting to access the bus, effect sequential arbitration decisions and then cause a packet transfer to be implemented following each arbitration decision. The embodiment described herein allows a module to request a deferred arbitration decision while the current transfer is occurring.

[0031] Before describing the arbitration mechanism, the control signals of Figure 3 will be described:

[0032] **request** (req) ready to send data.

[0033] This is driven by an initiator module (IM) and is used to indicate that it is ready to transfer a request or element of a request across the interface. If this signal is asserted the request content and framing signals are valid.

[0034] Initiators indicate they have data to send by asserting request and expect a grant in this or subsequent cycles to complete the transfer.

[0035] **grant** (gnt) ready to accept data.

[0036] This is driven by a target module (TM) and is used by the target to indicate it is ready to accept data. A data or cell transfer across the interface occurs when the initiator is ready to send and the target is ready to accept, i.e. both request and grant are asserted at a rising clock edge.

[0037] Targets indicate they are able to accept data by asserting grant and expect a request in this or subsequent cycles to complete the transfer.

[0038] **end of packet** (eop) final cell of packet.

[0039] This driven by the initiator and indicates this is the final cell of a request packet.

[0040] **address** (add) the transaction target address.

[0041] This holds the address of the location within the target module the operation will occur on. It is field 63 in Figure 4.

[0042] **source** (src) source identifier.

[0043] Identifies the source of the transaction to the system. It allows the system (and target modules) to associate a series of transactions with a specific source of data. It is used as the destination address of a response packet for the associated request.

[0044] **response request** (r_req) indicates a response cell is available.

[0045] An initiator should only commence a transfer if it is ready to accept the response packet.

[0046] **response grant** (r_gnt) indicates a response cell may be accepted.

[0047] **response source** (r_src) is a copy of the source identification field which is used as the destination indicator of the response packet.

[0048] **next request** (a_req) indicates the module is ready to start the next or subsequent request packet.

[0049] **next grant** (a_gnt) indicates the system will be ready to accept the next or subsequent request packet on completion of the current packet.

[0050] **next address** (a_add) is the address of the target module for the next or subsequent request, being the dest byte 73 of Figure 4.

[0051] **next response request** (ar_req) indicates the module is ready to start the next or subsequent response packet.

[0052] **next response grant** (ar_gnt) indicates the system is able to accept the next or subsequent response on completion of the current.

[0053] **next response source** (ar_src) the destination of the next or subsequent response packet, being the SRC byte 99 of Figure 5.

[0054] The latter six signals (a_req, a_gnt, a_add, ar_req, ar_gnt, ar_src) are used to create a path from the module to the arbitration unit 22 which allows information on a later transfer to be dealt with whilst the current transfer is being implemented. Thus, a module can hide the latency associated with the arbitration function by supplying information on the later transfer to be considered whilst the current transfer is being implemented.

[0055] Figure 6 is a block diagram of an arbitration unit 22 and Figures 7a and 7b are timing diagrams illustrating an example of such a protocol showing the arbitration function considering the (n+2)th transfer while the nth transfer is occurring.

[0056] The arbitration mechanism will now be described with reference to Figure 6 and 7. Figure 6 is a schematic block diagram of the arbitration unit 22. It comprises decision making logic 13, a decision queue 115 having a plurality of decision locations 115a, transfer logic 19 and a control block 21 which controls the functions of the arbitration unit. A flow control mechanism 35 is arranged between the decision making logic 13 and the queue 115. The signals which go into and leave the arbitration unit 22 are also shown schematically in Figure 6 indicating the block to which they refer. The decision making logic 13 makes arbitration decisions according to any suitable arbitration algorithm as discussed earlier.

An arbitration decision is made responsive to receipt of an arbitration request `a_req` together with the destination byte 73 (`a_add`) of a packet generated in the functional module which made the arbitration request. Once the arbitration decision has been made, it is placed on a queue 115 in a decision location 115a. The flow control mechanism 35 only allows a decision to be made if there is capacity in the queue. If a decision has been made, a handshake control signal `a_gnt` is issued indicating that the system is committed to transfer that packet and that a subsequent arbitration request may be received. Each arbitration decision held in the queue 115 identifies the requesting module (`src 99`), the packet destination (destination byte 73 `a_add`) and any end of packet (`eop`) signal. The transfer logic 19 implements a transfer responsive to assertion of the request signal `req` of the source module and the grant signal `gnt` of the destination module. This may be needed multiple times depending on the number of cells in the packet and the bus width. The transfer logic 19 implements a transfer by assertion of a grant send signal (`gnt_send`) sent to the appropriate interface 6 causing the source module to put the packet onto the bus 15. The transfer logic then asserts a send signal (`snd`) which signals to the receiving module that it should accept the transfer currently on the bus. The packet transfer concludes when a cell with the `eop` signal asserted is present.

[0057] While the transfer logic 19 is implementing a transfer, an arbitration decision can be effected by the decision logic 13 for a later packet. Thus, from the module's point of view, it can request the transfer of a packet by assertion of the request signal `req` and simultaneously or a number of cycles later request arbitration for a subsequent packet ready for transfer. From the point of view of the

arbitration unit 22, it can deal with a transfer request and a request for an arbitration decision in each cycle, and in fact can make a number of sequential arbitration decisions which will be held in the queue 115 while the transfer is being effected. It is possible that a transfer of a packet for which an arbitration decision has been made will involve the assertion in multiple cycles of the request and grant signals, depending on the number of cells in the packet and the bus width.

[0058] Once a packet has been transferred, the decision relating to that packet is removed or retired from the queue. The queue additionally includes a mechanism to allow response packets to overtake request packets to avoid deadlock conditions where initiator modules cannot complete transactions because they are still awaiting return of a response packet.

[0059] Figure 7a illustrates the effective transfer of packets from a destination to a source module across the routing bus 15. The request signal req indicates that a function module has requested a transfer, and the grant signal gnt indicates that the destination module is able to receive the transfer. The address signal 63 (add) indicates the location within the target module to which the packet is addressed. That is, it is the part of the address illustrated in Figure 4 which identifies the destination location within the target module. The destination of the target module itself, indicated by the destination (dest) byte 73 forms the signal a_add in Figure 7b. Figure 7b also illustrates the arbitration request signal a_req indicating that a request for arbitration has been made by a functional module. The arbitration grant signal a_gnt is asserted by the arbitration unit 22 when each arbitration decision has been made. At time

to, an arbitration request `a_req` is made by a functional module and the destination byte 73 `a_add` of the address is supplied with the arbitration request. When the arbitration decision has been made at time `t1`, the arbitration grant signal `a_gnt` is asserted which acts as a handshake control to the functional module which issued the arbitration request to indicate that an arbitration decision has been made and that its packet is ready for transfer. The time between `t0` and `t1` is a decision cycle time. Once the decision has been made, it is queued in the queue 115 ready to be implemented as a transfer. In order to implement the transfer, the functional module raises the request signal `req` and supplies the destination byte 73 (`add`). When the grant signal `gnt` of the destination module is asserted at time `t3` the transfer can be implemented. During that transfer, the decision logic 13 has not been idle. A subsequent destination address for a second packet `REQ2` was supplied by the functional module at time `t4` while asserting the arbitration request `a_req`. The arbitration decision was made one decision cycle later and the arbitration grant signal asserted at time `t5`. That arbitration decision was placed in the queue whilst the transfer logic 19 was implementing the transfer of the first request `REQ1`. A subsequent request, `REQ3` was also the subject of an arbitration decision during the transfer time of the first request `REQ1`. That decision is also placed on the queue 15. By the time the arbitration decision for the third request has been implemented, the transfer mechanism is ready to transfer a subsequent packet. Thus, at time `t6` there are two decisions in the queue, `REQ2`, `REQ3`. `REQ1` has been transferred and `REQ2` is awaiting transfer from the queue. When the request signal `req` is next asserted by a functional module, at time `t6` and the subsequent grant signal from the

destination module is asserted, then the transfer of the second request REQ2 can take place. This can coincide with the making of a subsequent arbitration decision for a fourth request, REQ4. Thus, the latency of arbitration decisions can be hidden by overlapping with the transfer time.

[0060] It will be appreciated that a transfer may involve multiple assertions of the request signal req until a transfer with eop asserted is made. However, it is only necessary to make a single arbitration decision per packet.

[0061] Although the sequence has been described for module M1 as though it is an initiator module making a request to transmit a request packet, a similar sequence of events takes place for the transmission of response packets. This involves the insertion of a response request signal r_req, together with transmission of the src byte indicating the destination of the response packet (being the initiator module). A response grant signal r_gnt is asserted by the initiator module when it is ready to receive a response. These signals are all omitted from Figure 3 for the sake of clarity, but it will be understood that they extend from the target module interface to the arbitration unit 22.